In the Claims:

1. (Currently Amended) A method of converting a received client signal containing client data units into a frame-based transport signal at a higher bit rate, comprising the steps of:

creating successive payload sections each accommodating the same number of transport data units, each transport data unit being a byte, each transport data unit also being set either to one of the [[a]] received client data [[unit]] units or to a dummy data unit~~, wherein pairs of contiguous client data units from the received client signal are positioned in transport data units that are separated by at least one transport data unit containing a dummy data unit~~; and

creating successive frames of the transport signal by appending ancillary data to each payload section;

wherein the number of client data units carried by the payload section of each frame is within one client data unit of the actual number of client data units received during the duration of that frame.

2. (Original) A method as claimed in claim 1,

wherein the number of client data units received during the duration of a frame is a whole number; and

wherein the number of client data units carried by the payload section of each frame is exactly equal to said whole number.

3. (Original) A method as claimed in claim 2, wherein said whole number is less than the number of transport data units accommodated by the payload section of a frame.

4. (Original) A method as claimed in claim 1,

wherein the number of client data units received during the duration of a frame is a fractional number falling between two adjacent integers F_L and F_H; and

wherein the number of client data units carried by the payload section of each frame is equal to either F_L or F_H; and

wherein the ancillary data associated with the frame contains information related to the chosen number of client data units carried by the payload section in that frame.

5. (Original) A method as claimed in claim 4, wherein the integers F_L and F_H are functions of both the data unit rate of the client signal and the frame rate of the transport signal.

6. (Original) A method as claimed in claim 4, wherein the integer F_L is equal to the greatest integer which is less than or equal to the number of client data units received per second multiplied by the duration of a frame.

7. (Original) A method as claimed in claim 4, wherein the information related to the number of client data units carried by the payload section of a frame is redundantly encoded in the ancillary data associated with that frame.

8. (Original) A method as claimed in claim 4, further comprising the step of temporarily storing the received client data units in a buffer having a measurable fill level, wherein the number of client data units carried by the payload section of each frame is a function of the fill level of the buffer.

9. (Original) A method as claimed in claim 1, wherein client data units are substantially evenly distributed within the payload section of each frame of the transport signal.

10. (Currently Amended) A method as claimed in claim 1, wherein the first transport data unit in the payload section of each frame is set to a first client data unit and wherein the method further comprises the step of:

incrementing a counter each time a transport data unit is set either to [[a]] one of the client data [[unit]] units or to [[a]] the dummy data unit, by an amount equal to the number of client data units carried by the payload section of the current frame, wherein the counter has a modulus equal to the total number of transport data units accommodated by the payload section of each frame;

wherein each transport data unit other than the first one in the payload section of the current frame is set either to [[a]] the client data unit or to the dummy data unit as a function of the overflow status of the counter after performing the incrementing step for that transport data unit.

11. (Original) A method as claimed in claim 1, further comprising the steps of:

extracting a client clock signal from the client signal;

processing the client clock signal to produce a transport clock signal at a frequency of M/N times the frequency of the client clock signal, where M and N are integers; and

outputting the transport signal in synchronism with the transport clock signal.

12. (Original) A method as claimed in claim 11, wherein the values of M and N are transmitted as part of the ancillary data associated with at least one frame.

13. (Original) A method as claimed in claim 1, wherein the transport signal is compliant with the G.975 standard.

14. (Currently Amended) An article of manufacture, comprising:

a computer usable medium having computer readable program code means embodied therein for causing the conversion of a received client signal containing client data units into a frame-based transport signal at a higher bit rate, the computer readable program code means in said article of manufacture comprising:

computer readable program code for causing a computer to create successive payload sections each accommodating the same number of transport data units, each transport data unit being a byte and each transport data unit being set either to [[a]] one of the received client data [[unit]] units or to a dummy data unit~~, wherein pairs of contiguous client data units from the received client signal are positioned in transport data units that are separated by at least one transport data unit containing a dummy data unit~~; and

computer readable program code for causing a computer to create successive frames of the transport signal by appending ancillary data to each payload section;

wherein the number of client data units carried by the payload section of each frame is within one client data unit of the actual number of client data units received during the duration of that frame.

15. (Currently Amended) A system for converting a received client signal containing client data units into a frame-based transport signal, comprising:

a first processing module adapted to output transport data units, each of which is set either to one of the received client data units or to a dummy data unit, wherein each transport data unit is a byte, ~~wherein pairs of contiguous client data units from the received client signal are positioned in transport data units that are separated by at least one transport data unit containing a dummy data unit,~~ the first processing module being further adapted to ensure that the number of client data units output during each frame is within one client data unit of the number of client data units received during the duration of that frame; and

a second processing module connected to the first processing module and adapted to create successive payload sections each accommodating the same number of transport data units received from the first processing module, the second processing module being further adapted to create successive frames of the transport signal by appending an ancillary data section to each payload section and outputting the data units in the ancillary section and the payload section faster than the rate at which the client signal is received.

16. (Original) A system as claimed in claim 15, further comprising:

a clock recovery circuit, for recovering a clock which is synchronous with the received client signal;

a clock processing module connected to the clock recovery circuit and to the first processing module, for generating a gating signal which indicates to the first processing module when to output a transport data unit; and

a clock frequency multiplication circuit connected to the clock recovery module and to the second processing module, for generating the faster clock signal used by the second processing module in outputting the transport signal.

17. (Original) A system as claimed in claim 16, wherein the clock recovery circuit comprises an M/N-multiplying phase-locked loop, wherein M and N are integers.

18. (Original) A system as claimed in claim 17, wherein the values of M and N are supplied by the first processing module via a control line.

19. (Original) A system as claimed in claim 16, further comprising:

a memory element connected to the first processing module and to the clock recovery circuit, for storing client data units under control of the clock recovery circuit and for outputting client data units to the first processing module under control of the first processing module.

20. (Original) A system as claimed in claim 19, wherein the memory element is adapted to produce a control signal indicative of whether its capacity has reached a pre-determined fill level.

21. (Original) A system as claimed in claim 20, wherein the memory element supplies said control signal to the first processing module, wherein the first processing module is adapted to decide how many client data units to output during each frame as a function of the value of said control signal.

22. (Original) A system as claimed in claim 15, wherein the first processing module is adapted to output client data units at substantially periodic intervals.

23. (Currently Amended) A system as claimed in claim 22, wherein the first processing module is adapted to ensure that the first transport data unit in the payload section of each frame is set to a client data unit and wherein the first processing module is further adapted to:

increment a counter each time a transport data unit is set either to [[a]] <u>one of the</u> client data [[unit]] <u>units</u> or to [[a]] <u>the</u> dummy data unit, by an amount equal to the number of client data units carried by the payload section of the current frame, wherein the counter has a modulus equal to the total number of transport data units accommodated by the payload section of each frame;

wherein each transport data unit other than the first one in the payload section of the current frame is set either to [[a]] one of the client data [[unit]] units or to [[a]] the dummy data unit as a function of the overflow status of the counter after performing the incrementing step for that transport data unit.

24. (Currently Amended) A method of converting a received transport signal into a client signal, said transport signal comprising a series of frames each containing a payload section and an ancillary data section appended to the payload section, wherein each payload section comprises a plurality of transport data units each of which is either a client data unit or a dummy data unit, the method comprising the steps of:

determining whether each transport data unit in each frame of the transport signal is [[a]] one of the client data [[unit]] units or a dummy data unit, wherein each dummy data unit is a byte; and

outputting to a buffer only those transport data units determined to be client data units~~, wherein the transport signal includes a pair of transport data units containing client data units that are separated by at least one transport data unit containing a dummy data unit, said pair of transport data units containing client data units being output to the buffer consecutively such that they are positioned contiguously in the client signal derived from the received transport signal~~.

25. (Original) A method as claimed in claim 24, further comprising the step of:

creating the client signal by reading from the buffer at a rate lower than that of the transport signal.

26. (Currently Amended) A method as claimed in claim 25, the method further comprising the steps of:

treating the first transport data unit in the payload section of each frame as [[a]] one of the client data [[unit]] units; and

incrementing a counter each time a transport data unit is processed, by an amount equal to the number of client data units carried by the payload section of the current frame, wherein the counter has a modulus equal to the total number of transport data units accommodated by the payload section of each frame;

wherein each transport data unit other than the first one in the payload section of the current frame is determined to be either [[a]] <u>one of the</u> client data [[unit]] <u>units</u> or [[a]] <u>the</u> dummy data unit as a function of the overflow status of the counter after performing the incrementing step for that transport data unit.

27. (Currently Amended) A method as claimed in claim 26, wherein the number of <u>client</u> data units carried by the payload section of the current transport frame is determined by reading information contained in the ancillary data section associated with that frame.

28. (Original) A method as claimed in claim 24, further comprising the steps of:

determining the values of the numerator and denominator of a clock frequency multiplier ratio;

extracting a transport clock signal from the transport signal;

processing the transport clock signal to produce a client clock signal at a frequency equal to the frequency of the transport clock signal times the clock frequency multiplier ratio; and

outputting the client signal in synchronism with the client clock signal.

29. (Original) A method as claimed in claim 28, wherein the values of the numerator and denominator are determined from the ancillary data section associated with at least one frame.

30. (Currently Amended) An article of manufacture, comprising:

a computer usable medium having computer readable program code means embodied therein for causing the conversion <u>of</u> a received transport signal into a client signal, said transport signal comprising a series of frames each containing a payload section and an ancillary data section appended to the payload section, wherein each payload section comprises a plurality of transport data units each of which is either a client data unit or a dummy data unit, the computer readable program code means in said article of manufacture comprising:

computer readable program code for causing a computer to determine whether each transport data unit in each frame of the transport signal is [[a]] <u>one of the</u> client data [[unit]] <u>units</u> or [[a]] <u>the</u> dummy data unit<u>, wherein each transport data unit is a byte</u>; and

computer readable program code for causing a computer to output to a buffer only those transport data units determined to be client data units~~, wherein the transport signal includes a pair of transport data units containing client data units that are separated by at least one transport data unit containing a dummy data unit, said pair of transport data units containing client data units being output to the buffer consecutively such that they are positioned contiguously in the client signal derived from the received transport signal~~.

31. (Currently Amended) A system for converting a transport signal into a client signal, said transport signal comprising a series of frames each containing a payload section and an ancillary data section, wherein the payload section carries transport data units each of which can be a client data unit or a dummy data unit, the system comprising:

a first processing module adapted to locate the payload section of each received frame and to output the transport data units in the payload section of each frame; and

a second processing module connected to the first processing module, the second processing module being adapted to:

determine whether each transport data unit in each frame of the transport signal is [[a]] one of the client data [[unit]] units or [[a]] the dummy data unit, wherein each transport data unit is a byte; and

output to a buffer only those transport data units found to be client data units~~, wherein the transport signal includes a pair of transport data units containing client data units that are separated by at least one transport data unit containing a dummy data unit, said pair of transport data units containing client data units being output to the buffer consecutively such that they are positioned contiguously in the client signal derived from the received transport signal~~.

32. (Currently Amended) A system as claimed in claim 31, further comprising:

[[a]] the buffer connected to the second processing module, for temporarily storing the client data units output by the second processing module and for outputting its contents at a controllable rate;

a clock recovery circuit, for recovering a clock which is synchronous with the transport signal; and

a clock frequency multiplication circuit connected to the clock recovery circuit and to the buffer, the clock frequency multiplication circuit being adapted to produce a client clock signal at a rate which is less than that of the transport signal and to provide the client clock signal to the buffer so as to control the output rate thereof.

33. (Original) A system as claimed in claim 32, wherein the ancillary data section associated with at least one frame contains a numerator and a denominator and wherein the first processing module is adapted to read the numerator and the denominator and to provide these to the clock frequency multiplication circuit.

34. (Currently Amended) A system as claimed in claim 33, wherein the clock frequency multiplication circuit is [[an]] <u>a</u> numerator/denominator-multiplying phase-locked loop.

35. (Original) A system as claimed in claim 31, wherein the second processing module is adapted to output client data units at substantially periodic intervals.

36. (Currently Amended) A system as claimed in claim 35, wherein the second processing module is adapted to ensure that the first transport data unit in the payload section of each frame is considered to be a client data unit and wherein the first processing module is further adapted to:

increment a counter each time a transport data unit is processed, by an amount equal to the number of client data units carried by the payload section of the current frame, wherein the counter has a modulus equal to the total number of transport data units accommodated by the payload section of each frame;

wherein each transport data unit other than the first one in the payload section of the current frame is considered to be either [[a]] <u>one of the</u> client data [[unit]] <u>units</u> or [[a]] <u>the</u> dummy data unit as a function of the overflow status of the counter after performing the incrementing step for that transport data unit.

37. (Original) A system as claimed in claim 36, wherein the first processing module is further adapted to determine how many client data units are carried by the payload section of the current frame by reading information in the ancillary data section associated with that frame.

38. (Currently Amended) A method for converting a client signal containing client data units into a frame-based transport signal compatible with a transport network and for retrieving the client signal at another part of the network, comprising the steps of:

(A) at an ingress interface:

creating successive payload sections each accommodating the same number of transport data units, each transport data unit being set either to a received client data unit or to a dummy data unit, wherein each transport data unit is a byte~~, wherein pairs of contiguous client data units from the received client signal are positioned in transport data units that are separated by at least one transport data unit containing a dummy data unit~~;

creating successive frames of the transport signal by appending ancillary data to each payload section, wherein the number of client data units carried by the payload section of each frame is within one client data unit of the actual number of client data units received during the duration of that frame; and

sending the frames into the transport network towards an egress interface at a data unit rate higher than that of the client signal; and

(B) at the egress interface:

determining whether each transport data unit in each frame is [[a]] one of the client data [[unit]] units or [[a]] the dummy data unit;

outputting to a buffer only those transport data units determined to be client data units; and

reading from the buffer at the lower data unit rate, thereby to recover the original client signal.